

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Computer Science curriculum

Taavi Gilden

**Creating a Human Similarity Ratings Benchmark
Database for Artificial Neural Networks**

Bachelor's Thesis (9 ECTS)

Supervisors: Jaan Aru, PhD
Martin Hebart, PhD
Raul Vicente, PhD

Tartu 2015

Creating a Human Similarity Ratings Benchmark Database for Artificial Neural Networks

Abstract:

In recent years, there has been a growing interest in understanding the way in which humans encode and categorise visual images. One hope is that this knowledge can be translated into improvements in artificial agents that carry out image classification automatically. One approach for closing in on the code of human object categorisation is to understand the relational coding of different objects in terms of their similarity. However, due to the large number of possible objects and their even larger number of possible comparisons, classical experimental approaches are limited in building representational similarity spaces.

The goal of this project was to create a web-application, which could be accessed by a large number of people around the world, who could contribute to the goal of the project. The webpage allows subjects to rate natural images according to their perceived similarity. This yields a large and reliable dataset of image ratings that could be accessed and used by experimenters, but also by image classification experts worldwide. Based on the dataset a similarity matrix can be constructed. The webpage and the corresponding database will hopefully advance our understanding of human vision and may one day be used to improve artificial intelligence algorithms designed to categorise objects.

Keywords: representational similarity analysis, similarity ratings, similarity, multiple item arrangement

Inimteadmiste kasutamine, loomaks sarnasushinnangute andmebaasi tehisnärvivõrkude arendamiseks

Lühikokkuvõte:

Viimaste aastate jooksul on kasvanud huvi selle vastu, kuidas inimesed tajuvad ja kategoriseerivad pilte. Loodetakse, et seda informatsiooni saab kasutada selleks, et arendada ja täiustada tehislikke pildiklassifitseerimissüsteeme. Üks suund, kuidas mõista inimteadmist piltide klassifitseerimisel, on uurida erinevate objektide sarnasusi. Kuid suure hulga võimalikke objektide tõttu, mis tekitaksid veel suurema hulga võimalikke võrdlusi, on klassikaliste eksperimentide võimalused sarnasusruumide konstrueerimiseks piiratud.

Antud projekti eesmärgiks oli luua veebirakendus, millele pääseks ligi suur hulk inimesi üle maailma, kes saaksid panustada antud projekti sihti. Veebirakendus võimaldab kasutajatel hinnata loomulikke pilte tajutava sarnasuse põhjal. See toodaks suure ja usaldusväärse andmehulga piltide sarnasustest, millele pääseksid ligi ja mida saaksid kasutada nii teadlased kui ka pildiklassifitseerimise eksperdid üle maailma. Kogutud andmete põhjal koostatakse sarnasusmaatriks. Antud rakendus ning sellega kaasnev andmebaas aitavad meil loodetavasti paremini mõista inimnägemist. Samuti võib antud andmehulk kasuks tulla objektide kategoriseerimiseks loodud tehisintellekti algoritmide arendamisel.

Võtmesõnad: kujutatav sarnasus analüüs, sarnasus hinnangud, sarnasus, mitme eseme asetus

Table of Contents

Introduction	6
1 Research methods.....	7
1.1 Representational similarity analysis	7
1.2 RSA as a common currency for behaviour, neural data and models	8
1.3 Motivation for a webpage for collecting similarity ratings	9
2 Webpage.....	10
2.1 Code and algorithms	10
2.2 Multiple item arrangement method.....	10
2.3 Multiple item arrangement algorithm	12
2.3.1 Estimating RDM from multiple item-subset arrangements.....	13
2.3.2 Lift-the-weakest algorithm	14
2.3.3 Example	16
2.4 Database.....	25
2.4.1 Users table	25
2.4.2 Login attempts table	26
2.4.3 Images table.....	26
2.4.4 Arrangements table.....	26
2.4.5 RDM table	27
2.5 Functionality	27
2.5.1 Authentication	27
2.5.2 Registration.....	28
2.5.3 Application functionality	28
2.5.4 Accessing the resulting RDM.....	31
2.6 Technologies and languages	31
2.6.1 PHP	31
2.6.2 JavaScript	32
2.6.3 MySQL	32
2.6.4 HTML.....	33
2.6.5 CSS	33
3 Results	34
3.1 Subjects.....	34
3.2 Task.....	34
3.3 Results summary.....	34
4 Discussion	37
4.1 Necessity of the similarity ratings	37

4.2	Webpage cons and pros	37
4.2.1	Future improvements	37
	Conclusion	40
	References	41
	Licence.....	43

Introduction

There has been a lot of progress in creating and training artificial neural networks over the last 5 years [1]. This has had a massive impact on artificial image classification. For example artificial neural networks are better at classifying traffic and road signs than humans are [1]. Nevertheless human knowledge still greatly exceeds these algorithms in many aspects [2]. To make improvements in these algorithms and machine vision in general, a comparison to human visions needs to be made. To achieve that a method is needed, which would help to relate human and machine vision.

Both of these visual systems, human and machine, are similar in a way that they are both hierarchical in their structure [1]. However, it could be that the systems are even more similar in their functionality. How could one directly compare human vision and artificial visual systems? One approach for linking the two would be similarity analysis.

Similarity analysis tries to quantify the similarities of different objects. Using representational similarity analysis (RSA), one can compare data from different sources, for example human similarity ratings, brain activation patterns, or activations of artificial neuronal networks [3]. The goal of the present thesis is to create a webpage for collecting data on how humans compare images. That data could then in the future be used for different research projects on finding the similarities between machine vision and human vision.

Until recently, RSA analyses have only been done with up to 100 images, which is quite a limited amount of data and may not be sufficient to draw strong conclusions: these image sets have been selected according to predefined categories and only represent a small fraction of possible object categories to choose from. To give much better benchmark of data for similarity analyses we use around 1400 images chosen from the most frequent concrete nouns in the English language, which gives the research community a much bigger and possibly more representative dataset.

1 Research methods

1.1 Representational similarity analysis

The research method used to collect human knowledge using the web-application is a new experimental and data-analytical framework called representational similarity analysis (RSA) [3]. It requires computing representational dissimilarity matrices (RDMs). RDMs are square matrices, where the number of columns or rows corresponds to the number of experimental stimuli. The matrix is symmetric about a diagonal of zeros. A RDM contains data for each pair of images. The data located in the matrix cells reflects the dissimilarity between images that the cell corresponds to, as shown in Figure 1. The data about the dissimilarity can come from different sources: one could look at the similarity of the activity patterns in neural data across a number of neurons, do the same in artificial neural networks, or could let human subjects directly rate the similarities between images.

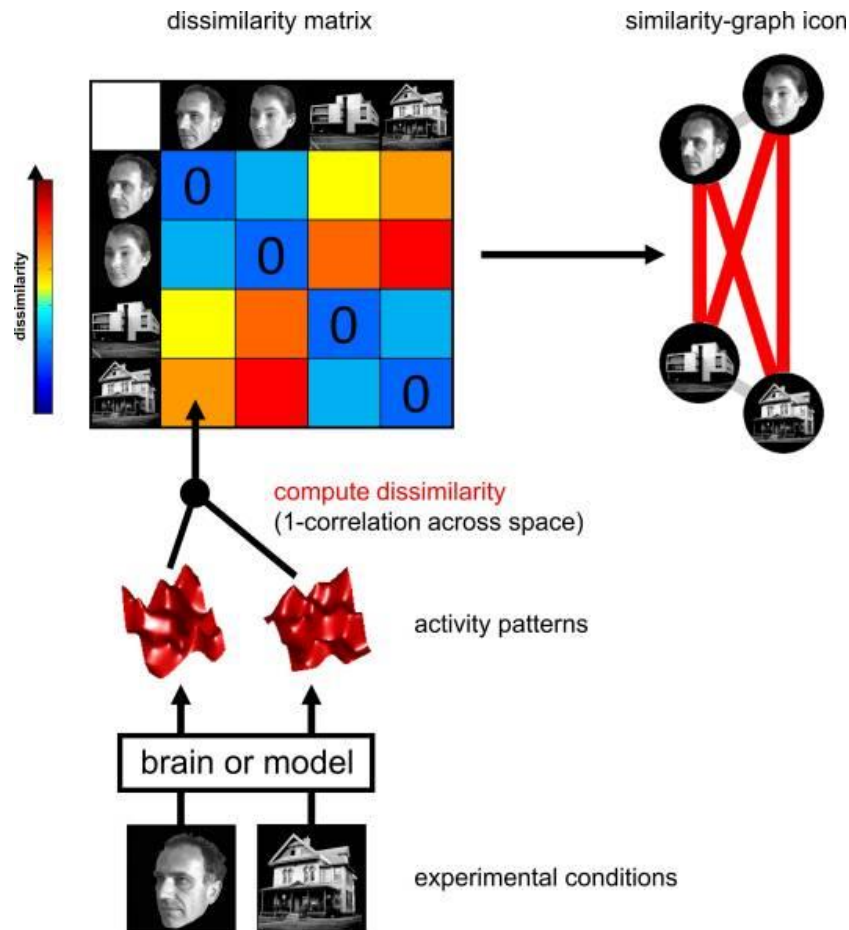


Figure 1: Computation of representational dissimilarity matrix [3]

The web-application developed in the present thesis produces these representational similarity matrices based on how humans compare the images that are given to them (explained below). Values that enter the cells of a RDM via the application correspond to the rated distance between images. The RDM contains similarity data for every image pair in the matrix.

1.2 RSA as a common currency for behaviour, neural data and models

It is possible to categorise the major methodologies of systems neuroscience as brain-activity measurement, behavioural measurement and computational modelling. The problem would be to quantify the data and relate the data collected from each of the different methods. Representational similarity analysis acts as a link between different representations. [3]

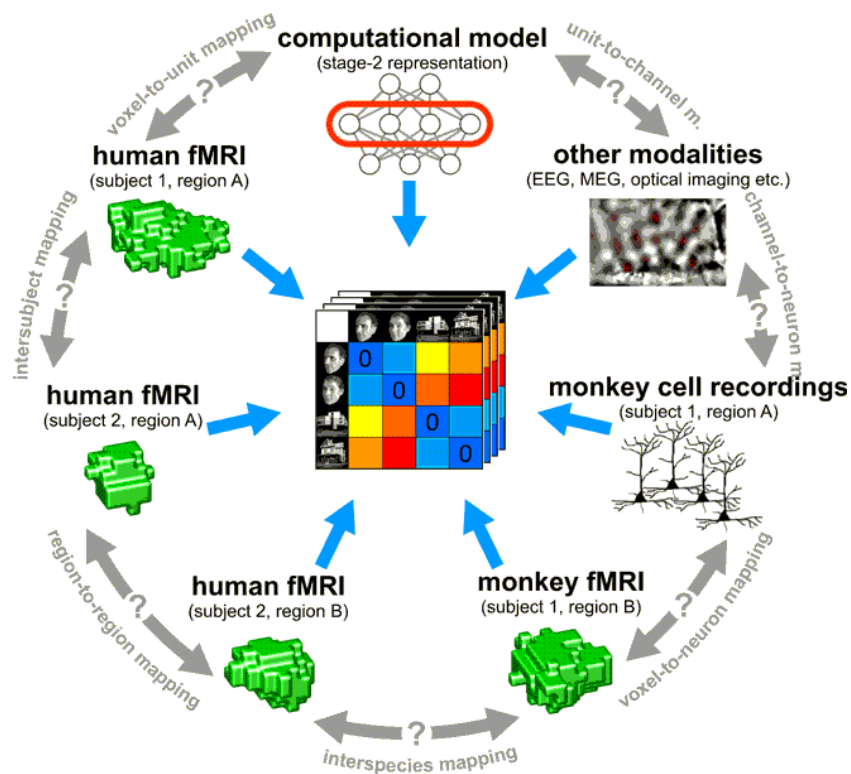


Figure 2: RSA allows to relate data from different sources [3]

Figure 2 illustrates the different representations that can be related using the RDM. It allows us to quantitatively relate the brain-activity measurement of different species, subjects, regions and modalities. It also makes it possible to evaluate, compare and integrate computational models of brain information into the data analysis of brain activity. Additionally, it allows to relate brain-activity measurements to behavioural data. Representational similarity analysis synergises well with experimental designs, where an experiment can address multiple questions related to neuroscience. [3]

1.3 Motivation for a webpage for collecting similarity ratings

The motivation for the webpage would be to make it simple for people to contribute to an advancement of neuroscience on the one hand, and potentially artificial neural networks on the other hand, without actually having to carry out experiments. When Nikolaus Kriegeskorte and Marieke Mur collected similarity ratings, they had 4 people to participate in the experiment [4]. It consisted of two 45-minute sessions on separate days [4]. The subjects had to physically go to the research lab to give similarity ratings, and the rating could only be given for a limited time, which means the data for each pair is also limited. Due to the time limitation not all pairs could be rated enough to provide a proper understanding on their dissimilarity.

The webpage allows to collect data from people all over the world and the amount of subjects is not limited. In addition the webpage removes the time limitation from the subjects since they can rate images whenever they have time to do so. The webpage is a simple tool for collecting data about human vision in the form of similarity ratings. Collecting data on such a large item set and from more subjects has the potential to provide much better benchmark data for similarity analysis.

2 Webpage

2.1 Code and algorithms

The webpage consists of a typical authentication [5] that is done using PHP [6] and JavaScript [7]. Users can create an account to register themselves to the website and use the information to authenticate themselves. When information is inserted to the registration page, the field values are checked for correctness by a JavaScript function. The password is hashed using a JavaScript implementation of a SHA-512 hashing algorithms [8]. A random salt is also generated for the hashing process. Both the hashed password and the generated salt are saved to the database. For the authentication process the username and password are checked. The password inserted to the authentication form is hashed with the same algorithm using the same salt and the hashes are compared to make sure the entered password would be equal to the real password. In addition the failed login attempts are stored and monitored for each user to prevent guessing the passwords by trial and error. When the authentication is successful a secure PHP session is made for the user and the user is directed to the main page of the application.

A PHP function exists in the code to initialise the image database. Its purpose is to populate the image database with the correct values for images stored in the server. It expects the images to be divided into different folders each representing the categories of the images it contains. It iterates over the given path the images are in and stores image location for each image to the database. The category values is taken from the folder name the images are in.

The main page of the application contains the arena and the images. The images are evenly placed around the arena in a circle using JavaScript. jQuery UI [9] is used to make the HTML [10] element representing the arena able to accept images dropped on it. In addition the images places around the arena are similarly given the ability to be dragged around the screen. Functions for changing the image size to present a larger preview of the image to the user when hovering over the image are also present. jQuery UI is also used to show the tooltips for each picture representing the category the images belongs to.

Ajax [11] is used for communication with the server.

2.2 Multiple item arrangement method

The key of the present web-application is the multi item arrangement method together with the algorithm for inverse multi-dimensional scaling [4]. This is the algorithm that firstly decides, which images to present to the subject and secondly in the background combines the

information for computing the RDM. Also, implementing this algorithm took the main share of the time invested into the present thesis. Hence, next the algorithm is described in full detail.

With the multi-arrangement method, a user can arrange multiple item subsets in different trials in a low-dimensional, in our case two-dimensional space, which is also shown in Figure 3. The subsets are adaptively designed as described in Section 2.3. The dissimilarity information is taken from the distances between items. The items that have a smaller distance between them are interpreted as more similar than the items that are placed farther apart. [4]

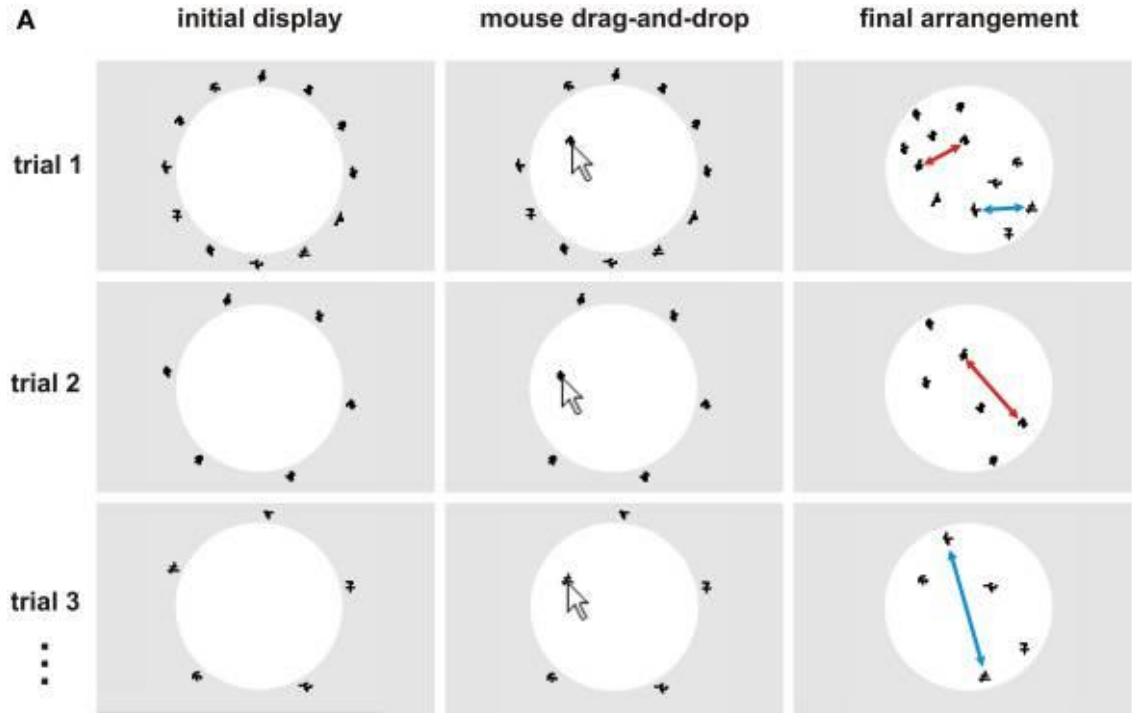


Figure 3: Multiple item arrangement method for collecting subject's dissimilarity judgements [4]

Since each trial in multiple item arrangement provides a partial representational dissimilarity matrix, it is required to combine that information into a single RDM. The process and the algorithm are described in Section 2.3.

2.3 Multiple item arrangement algorithm

The multiple item arrangement algorithm [4] is used to cycle through images from the database and produce representational similarity matrices from users' arrangements.

The applications works on three different kinds of data.

1. *Arrangement*: contains the positions of all the images after the user has finished arranging them on the arena.
2. *Dissimilarity-evidence matrix*: is calculated based on the arrangements. It is individual for each subject and it contains the sum of evidence weights (explained below) for each pair over all the arrangements the user has completed. In simple terms, this matrix contains information about whether one has enough evidence about a particular dissimilarity estimate.
3. *Representational dissimilarity matrix (RDM)*: is the end result, which could be used to compare the human visual system to other research modalities, for example artificial neural networks.

The following algorithm represents the method described in Section 2.2.

1. Present the set of images to the user that is supposed to be arranged.
 - a. If the user is on the first trial, he is presented 60 random images each of which is from a different category.
 - b. If the user is not on the first trial, we select images using the lift-the-weakest algorithm [4] explained below.
2. Estimate the representational dissimilarity matrix for the user, explained in Section 2.3.1.
3. If user's times is up or he has reached a criterion with his dissimilarity-evidence matrix we terminate.
 - a. If user's time is up, his RDM is saved to the database and the process can be resumed when the user returns.
 - b. We check if all the values in the dissimilarity-evidence matrix have reached a dissimilarity evidence of at least 0.5. That means if the evidence weight exceeds 0.5 the utility reaches its maximum. The algorithm then focuses on the items that have an evidence weight under 0.5 to lift the ones we have less evidence for.
4. Design the item-subset for the next trial using the lift-the-weakest algorithm, which is based on current dissimilarity evidences.
5. The subject is asked to arrange the item-subset presented to him.
6. The algorithm returns to Step 2.

2.3.1 Estimating RDM from multiple item-subset arrangements

Estimating RDM from multiple item-subset arrangement [4] is used in the Step 2 of the Section 2.3 algorithm. The algorithm is for estimating the representational dissimilarity matrices based on all the arrangements that the user has done previously. It analyses the arrangements from the database and scales them according to the following algorithm.

1. Check if user is on the first trial or not.
 - a. If user is on the first trial, then the algorithm generates the representational dissimilarity matrix based on the one arrangement the user has done. The algorithm goes over all the pairs in the arrangement and finds the value of distance between the images in the selected pair divided by the arena size. The value is then inserted into the correct location in the RDM. The empty values in the RDM are replaced with a constant of a relatively small value, in the present case with the value of PI divided by 10. The values are scaled, so that the root mean square of the values that are not PI divided by 10 would be equal to 1. The algorithm sends the RDM to the database and proceeds to Step 3 on the main algorithm.
 - b. If the user is not on the first trial the algorithm takes the last calculated representational matrix from the database and proceeds on that with the following steps.
2. In case of 1.b, take all the arrangements the user has previously arranged from the database and for every pair calculate the value of distance between the pair elements divided by the arena size. After that scale these values so that for each arrangement the root mean square of its pairs would be equal to the root mean square of the same pairs in the RDM.
3. Iterate over the RDM and replace the values that we have collected information about in each trial with the dissimilarity estimate, shown in Equation (1).

$$\text{dissimilarity estimate} = \frac{\sum_{ti} (\text{scaled distance}_{ti} * \text{unscaled distance}_{ti}^2)}{\sum_{ti} (\text{unscaled distance}_{ti}^2)} \quad (1)$$

The sum is calculated by iterating over every trial that contains the pair, ti represents the trial index. The scaled distance is taken from Step 2. The unscaled distance is also taken from Step 2, it represents the value of the distance divided by arena size before the scaling process. The values are summed across all arrangements for each pair.

4. The values of Step 3 are scaled, so that the root mean square of the values that are not PI divided by 10 would be equal to 1.
5. We take the RDM that we found in Step 4 and subtract the RDM that we had as input in Step 2 (i.e. the RDM from the previous trial). Find the root mean square of the representational dissimilarity matrix produced by the subtraction process and see if it is close to 0.
6. The algorithm goes back to Step 2 until the root mean square found in Step 5 is smaller than 0.00001 after which the algorithm finishes and saves the produced RDM to the database.

2.3.2 Lift-the-weakest algorithm

Lift-the-weakest algorithm [4] is used in the Step 4 of the Section 2.3 algorithm. The purpose of this algorithm is to find the item subset to be presented to the user in a subsequent trial. It takes into account the previous arrangements and finds the most optimal subset.

1. Take all the arrangements the user has arranged from the database and find the dissimilarity-evidence matrix. It contains the sum of evidence weights for each pair over all of the arrangements. Empty squares in the matrix are filled with PI divided by 10 and squared.
2. Find the pair that has the lowest dissimilarity evidence that we have actual dissimilarity information for.
3. The pair will initialise the item subset that the algorithm will design for the next trial. We also initialise the value of current trial efficiency.
4. We start adding elements to the current subset that are not already in it and for each subset we calculate the trial efficiency. By trying out every element we try to maximise the trial efficiency. If we have gone over all the unused items we return the subset that resulted in the best trial efficiency and update the current trial efficiency value to the new current best.
 - a. Trial efficiency is calculated based on trial benefit and trial cost. To calculate the trial efficiency we first need to calculate evidence utility for each pair in the subset.

- b. Evidence utility, shown in Equation (2) is based on evidence weight of the pair we are calculating it for. The weight is taken from the dissimilarity-evidence matrix.

$$evidence\ utility(w) = 1 - e^{-w \cdot d} \quad (2)$$

In the equation, w represents the evidence weight and d is a constant value of 10. Using the evidence utility, we calculate trial benefit.

- c. Trial benefit represents the total gain in evidence utility. It shows the amount of utility we gain if we present this subset to the user. For this we need to calculate the evidence utility sum before this trial and after the trial. To calculate the evidence utility sum before the trial we calculate the evidence utility for each pair in the current subset according to the Step 4.b and add them up. To calculate the evidence utility sum after the trial we use the same method, but we change the evidence weight of each pair. The evidence weight is now calculated by finding the largest distance in the subset and dividing each distance by that value and the result is squared to that new value we also add the weight of the pair from the dissimilarity-evidence matrix specified in Step 4.b. We do that for each pair in the subset and adding up all the values. Trial benefit is simply evidence utility sum before trial subtracted from the value of evidence utility sum after trial.
- d. Trial cost, shown in Equation (3) represents the time it takes to complete the trial.

$$trial\ cost = n^{1.5} \quad (3)$$

In the equation n is the number of items in a subset.

- e. Trial efficiency is trial benefit divided by trial cost.
5. The algorithm adds elements to the subset until the trial efficiency of the newest subset is lower than the trial efficiency of the subset before adding the last item. In other words the algorithm works until the trial efficiency is decreased after adding an element. After that the algorithm finishes and returns the subset to be presented to the user in the next trial.

It should be pointed out that when calculating the evidence weights for each pair in a single trial, we use the minimum value of 0.04. For closely placed images the placement noise starts dominating and estimating the evidence weight would be unreliable [4]. That means for example with a distance of 0.1 the evidence weight would be 0.01, which would require 100 trials to match the maximum single trial evidence weight of 1. So it is made sure that for distances below 0.2, which is about one fifth of the arena size, the evidence weight is 0.04. That requires approximately 25 trials to add up to the maximum single trial evidence weight of 1.

2.3.3 Example

The example shows the work of the algorithm described in Section 2.3. In this example let us assume that the database has a total of 5 images, which are i_1 , i_2 , i_3 , i_4 , i_5 . On the first trial we choose 3 images randomly. Let them be i_1 , i_3 and i_4 , in addition let the size of the arena be 600 pixels.

User arranges these 3 images on the arena based on their similarities and the result is an arrangement shown in Table 1.

Table 1: Distances between arranged elements on trial 1

	i_1	i_2	i_3	i_4	i_5
i_1	0		200	100	
i_2		0			
i_3	200		0	300	
i_4	100		300	0	
i_5					0

Table 2 shows the RDM that is calculated from the data of Table 1. In there we have divided all the distances by the size of the arena, values that are under 0.2 are valued to 0.2 and the empty cells are filled with a constant of $\pi/10$.

Table 2: Estimated RDM after the first trial

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.314	0.333	0.200	0.314
i_2	0.314	0	0.314	0.314	0.314
i_3	0.333	0.314	0	0.500	0.314
i_4	0.200	0.314	0.500	0	0.314
i_5	0.314	0.314	0.314	0.314	0

Before sending the RDM to the database we scale the values in the RDM that are not $\pi/10$ so that their root mean square would be equal to 1. The root mean square of the current values is equal to 0.401, which means that the scaling factor is 2.494 and the scaled RDM is calculated in Table 3.

Table 3: RDM scaled so that the root mean square equals to 1

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.314	0.831	0.499	0.314
i_2	0.314	0	0.314	0.314	0.314
i_3	0.831	0.314	0	1.247	0.314
i_4	0.499	0.314	1.247	0	0.314
i_5	0.314	0.314	0.314	0.314	0

After that the dissimilarity-evidence matrix, shown in Table 4, is calculated. On that matrix we check if all the values are over 0.5 and the algorithm should finish. We also design an item subset for the next trial based on that matrix.

Table 4: Dissimilarity-evidence matrix on trial 1

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.099	0.111	0.04	0.099
i_2	0.099	0	0.099	0.099	0.099
i_3	0.111	0.099	0	0.250	0.099
i_4	0.04	0.099	0.250	0	0.099
i_5	0.099	0.099	0.099	0.099	0

Lift-the-weakest algorithm selects two items from the ones that do not have the value of $(\pi/10)^2$. The pair that has the lowest dissimilarity evidence value is selected as the initial pair. In the current case that pair would be i_1 and i_4 , additionally we set the currently best trial efficiency to 0.

Lift-the-weakest algorithm starts choosing the best item to add to that subset, which would maximise the trial efficiency. To calculate the evidence utility sum before the trial we use the weight value from Table 4 and for each pair in the subset we calculate it according to Equation (2) located in Section 2.3.2. The largest distance is found from Table 1. The evidence utility

sum after the trial is calculated by finding the new evidence weight for the pair according to the largest distance in the subset and adding to it the evidence weight of the pair taken from Table 4. For example with i_1 and i_4 the new evidence weight would be $(100/100)^2$ and to that we add 0.04. With the value of 1.04 we calculate the evidence utility for that pair according to Equation (2). The trial benefit is the evidence utility sum before the trial subtracted from the evidence utility sum after trial. The trial cost is calculated as shown in Equation (3). Trial efficiency is trial benefit divided by trial cost. All the calculation results are shown in Table 5.

Table 5: Iteration 1 for finding the optimal item

Item subset	Current best trial efficiency	Evidence utility sum before the trial	Largest distance in the subset	Evidence utility sum after the trial	Trial benefit	Trial cost	Trial efficiency
$\{i_1, i_4, i_2\}$	0	1.587	100	3.000	1.413	5.196	0.272
$\{i_1, i_4, i_3\}$	0.272	1.918	300	2.861	0.943	5.196	0.181
$\{i_1, i_4, i_5\}$	0.272	1.587	100	3.000	1.413	5.196	0.272

The algorithm chooses the subset of i_1, i_2, i_4 , because the subset of i_1, i_4, i_5 did not produce a higher trial efficiency than the current best, which was 0.272. The new currently best trial efficiency is now 0.272 and the algorithm tries to add another item to the subset in the next iteration shown in Table 6. The calculations are done similarly to the iteration 1 shown in Table 5.

Table 6: Iteration 2 for finding the optimal item

Item subset	Current best trial efficiency	Evidence utility sum before the trial	Largest distance in the subset	Evidence utility sum after the trial	Trial benefit	Trial cost	Trial efficiency
$\{i_1, i_4, i_2, i_3\}$	0.272	3.803	300	5.840	2.037	8.000	0.255
$\{i_1, i_4, i_2, i_5\}$	0.272	3.472	100	6.000	2.528	8.000	0.316

The algorithm returns the subset of i_1, i_2, i_4, i_5 and tries to add another item in the next iteration shown in Table 7.

Table 7: Iteration 3 for finding the optimal item

Item subset	Current best trial efficiency	Evidence utility sum before the trial	Largest distance in the subset	Evidence utility sum after the trial	Trial benefit	Trial cost	Trial efficiency
$\{i_1, i_4, i_2, i_5, i_3\}$	0.316	6.317	300	9.811	3.494	11.180	0.313

The trial efficiency of the subset of i_1, i_4, i_2, i_5, i_3 was not higher than the current best, which was 0.316. That means that adding the last element of i_3 reduced the trial efficiency and the subset of i_1, i_2, i_4, i_5 is returned for the next trial.

The user arranges the next trial and the new distances are shown in Table 8.

Table 8: Distances between arranged elements on trial 2

	i_1	i_2	i_3	i_4	i_5
i_1	0	50		100	500
i_2	50	0		75	475
i_3			0		
i_4	100	75		0	400
i_5	500	475		400	0

The arrangement is stored in the database and a new estimate for the RDM is calculated. As a base we take the RDM displayed in Table 3. For calculating the new estimate for the RDM we need to take the previous arrangements. We need to find the scaling factor for each arrangement so that the root mean square of the arrangement would equal the root mean square of the same pairs in the current RDM taken from the database, in this case Table 3. First we find the unscaled values for the arrangement, for that we divide each value by the size of the arena, yet again the minimal value is 0.2, shown in Table 9.

Table 9: Unscaled values of trial 2 arrangement

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.200		0.200	0.833
i_2	0.200	0		0.200	0.792
i_3			0		
i_4	0.200	0.200		0	0.667
i_5	0.833	0.792		0.667	0

The root mean square of the values in Table 9 equals to 0.561 and the root mean square of the same values in the current RDM is 0.352, which makes the scaling factor equal to 0.627. The scaled values are shown in Table 10.

Table 10: Scaled values of trial 2 arrangement

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.125		0.125	0.522
i_2	0.125	0		0.125	0.497
i_3			0		
i_4	0.125	0.125		0	0.418
i_5	0.522	0.497		0.418	0

We also need to find the unscaled and scaled values for all the other previously completed arrangements, which means we have to take the arrangement evaluated in trial 1, shown in Table 1 and find the scaled and unscaled values for that as well. Unscaled values are shown in Table 11.

Table 11: Unscaled values of trial 1 arrangement

	i_1	i_2	i_3	i_4	i_5
i_1	0		0.333	0.200	
i_2		0			
i_3	0.333		0	0.500	
i_4	0.200		0.500	0	
i_5					0

The root mean square of these values is equal to 0.401 and the root mean square of the same pairs in the RDM taken from the database, shown in Table 3, is equal to 1, since the current RDM only consists of trial 1 arrangement. The scaling factor is 2.494 and the scaled values are shown in Table 12.

Table 12: Scaled values of trial 1 arrangement

	i_1	i_2	i_3	i_4	i_5
i_1	0		0.831	0.499	
i_2		0			
i_3	0.831		0	1.247	
i_4	0.499		1.247	0	
i_5					0

Now we can start putting together the new RDM. For that we need to find the dissimilarity estimate value, shown in Equation (1), for each pair. First we calculate the sum of squared unscaled values over every arrangement, shown in Table 13.

Table 13: Sums of unscaled values squared over all arrangements

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.040	0.111	0.080	0.694
i_2	0.040	0		0.040	0.627
i_3	0.111		0	0.250	
i_4	0.080	0.040	0.250	0	0.445
i_5	0.694	0.627		0.445	0

For next step we need to find the scaled values multiplied by unscaled value squared for each pair and then sum them up over all arrangements, shown in Table 14.

Table 14: Sums of unscaled values squared multiplied by scaled values over all arrangements

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.005	0.092	0.025	0.362
i_2	0.005	0		0.005	0.312
i_3	0.092		0	0.312	
i_4	0.025	0.005	0.312	0	0.186
i_5	0.362	0.312		0.186	0

For the next step we calculate the dissimilarity estimate value for each pair, which is Table 14 value divided by Table 13 value for each pair, results is shown in Table 15.

Table 15: Dissimilarity estimate values for each pair

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.125	0.829	0.313	0.522
i_2	0.125	0		0.125	0.498
i_3	0.829		0	1.248	
i_4	0.313	0.125	1.248	0	0.418
i_5	0.522	0.498		0.418	0

The dissimilarity estimate values in Table 15 are then scaled, so that their root mean square would be equal to 1. The root mean square of these values is equal to 0.619, which makes the scaling factor equal to 1.616. The scaled values are displayed in Table 16.

Table 16: Scaled dissimilarity estimate values for each pair

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.202	1.340	0.506	0.844
i_2	0.202	0		0.202	0.805
i_3	1.340		0	2.017	
i_4	0.506	0.202	2.017	0	0.675
i_5	0.844	0.805		0.675	0

Replace the scaled values of the dissimilarity estimates to the RDM we had in Table 3. As a results we get a new RDM shown in Table 17.

Table 17: New estimate for the RDM

	i_1	i_2	i_3	i_4	i_5
i_1	0	0.202	1.340	0.506	0.844
i_2	0.202	0	0.314	0.202	0.805
i_3	1.340	0.314	0	2.017	0.314
i_4	0.506	0.202	2.017	0	0.675
i_5	0.844	0.805	0.314	0.675	0

Now we take the RDM we had before replacing the new values, which is the one in Table 3, and subtract it from the new RDM estimate in Table 17, as a result we will get a RDM shown in Table 18.

Table 18: The results of the subtraction

	i_1	i_2	i_3	i_4	i_5
i_1	0	-0.112	0.509	0.007	0.530
i_2	-0.112	0	0	-0.112	0.491
i_3	0.509	0	0	0.770	0
i_4	0.007	-0.112	0.770	0	0.361
i_5	0.530	0.491	0	0.361	0

We calculate the root mean square of all the pairs from the matrix we got as a results of the subtraction, shown in Table 18. The root mean square is equal to 0.153, which is not less than 0.00001, so the algorithm uses the new RDM shown in Table 17 as an input to the loop, and starts the scaling progress all over again until the resulting matrix from the subtraction gives a root mean square less than 0.00001, after which the new RDM is returned and saved to the database.

The algorithm continues to the lift-the-weakest algorithm with a new dissimilarity-evidence matrix calculated by adding up all the evidence weights for each pair over all arrangements.

2.4 Database

From Figure 4, we can see that the database consists of 5 tables, which are the following:

1. users
2. login_attempts
3. images
4. arrangements
5. users_rdm

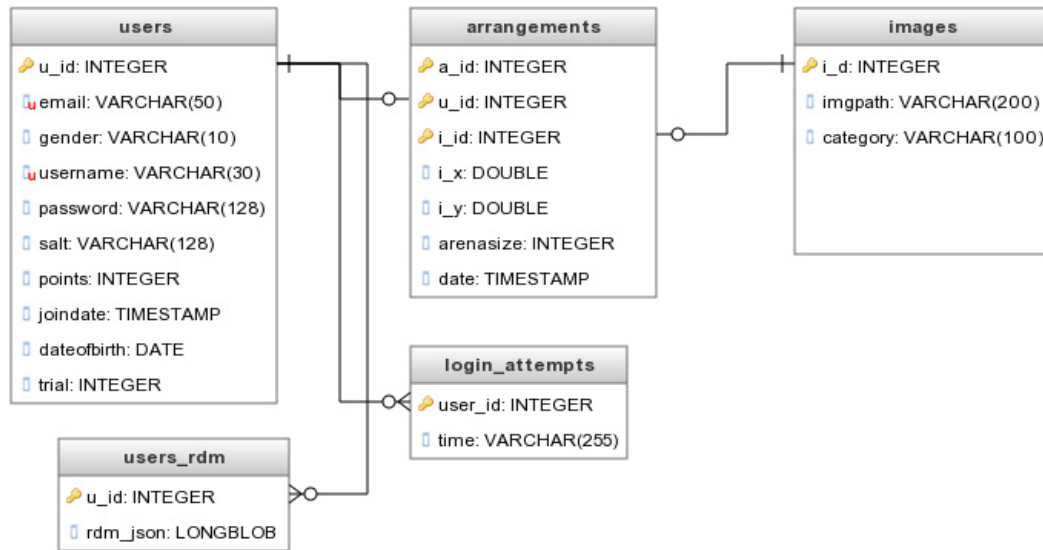


Figure 4: Unified Modelling Language (UML) diagram for the database

2.4.1 Users table

The users table stores all the data that the user inserts when creating an account. It consists of a unique id value, which is also a primary key for the table, from which we can identify the user. Id value is used in other tables as a foreign key. For each user we also store its e-mail address, gender, username, date of birth and the date on which they created their account. In addition a hashed password with the randomly generated salt are stored. We also store the trial number for each user that is used in algorithms to see how many trials the user has already completed. The column for storing points is also made for future implementations of the score system.

2.4.2 Login attempts table

In the login attempts table we store login attempts when user inserts false login information more than 5 times in short period of time. This is a security precaution to prevent someone from guessing a password by trial and error.

2.4.3 Images table

The images table is for storing all the data about the images that are used for the trials. Images are selected based on different categories. Categories are represented by 400 different nouns and for every noun 4 images were chosen. The nouns are picked by the following criteria:

1. 500 most frequently used nouns, from which 100 were removed
2. concrete, for example morning was removed
3. possible to show a representative example
4. if category contained at least 2 exemplars, then it was kicked, for example child contains boy and girl
5. body parts that were not the most prominent parts of a face or of limbs were removed

All the images are taken from ImageNet [12] image database. The categories that did not produce any eligible results were left out. In the end a total of 1328 images were chosen.

The table has a total of 3 columns, which are image id, image path and image category. First column being image id, contains a unique integer value to identify each image, the value is also automatically incremented. Image path represents the location of the image where it is stored on the server. Category shows in which of the previously stated categories the image belongs.

2.4.4 Arrangements table

Arrangements table is used to store all the arrangements that each user has made. The table consists of the arrangement id, the user id, the image id, the x and y coordinate of the image, the arena size and the time the arrangement was made.

The table has 3 primary keys and 3 foreign key constraints, which are arrangement id, user id and image id. Arrangement id is an integer value, which is calculated by subtracting user id from current time of the server, so it produces a unique number each time. User id is an integer value that shows to which user the arrangement belongs to. Image id identifies the image that is included in the arrangement. Coordinate values are integer values that show the location of the image at the time the arrangement was completed. Arena size value corresponds to the size of the arena the images were placed in pixels. Last column is for the time stamp that shows when the arrangement was sent to the database.

2.4.5 RDM table

Users RDM table stores all the representational dissimilarity matrices for each user. Only the upper triangle part of the matrix is stored since the matrix is symmetric with respect of the main diagonal.

The primary key and the foreign key for the table is user id. It is a unique integer value, which links the produced RDM to the user that completed the arrangements which values are stored in the matrix. RDM JSON column stores the serialised from of the matrix as a blob. The data is also compressed to make the string shorter, which also makes the data take up less space.

2.5 Functionality

2.5.1 Authentication

When first loading the webpage user is presented with the authentication form, shown in Figure 5. The form requires the user to enter a valid e-mail address and a password.

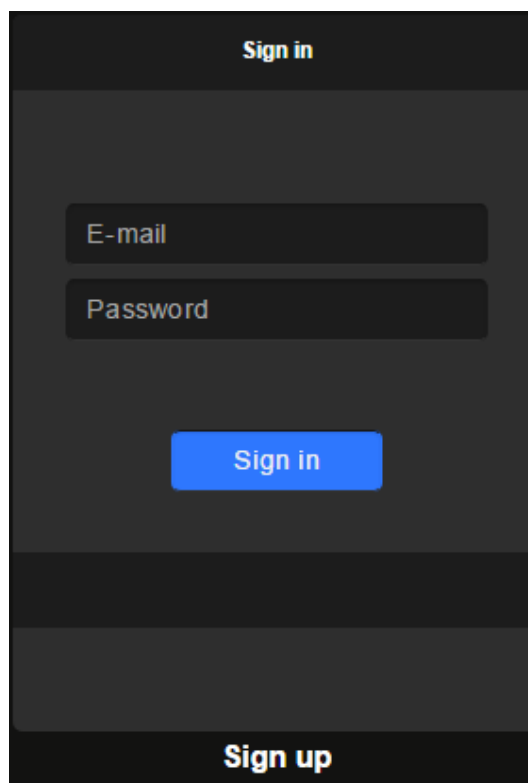
A screenshot of a mobile application's authentication form. The form has a dark background. At the top, there is a header bar with the text "Sign in" in white. Below this, there are two input fields: "E-mail" and "Password", both with white text. Below the input fields is a blue button with the text "Sign in" in white. At the bottom of the form, there is a footer bar with the text "Sign up" in white.

Figure 5: Authentication form

The same information given in the registration process should be entered. When user is done inserting the required login information the button “Sign in” can be clicked after which the server authenticates the user. If authentication is successful the user will be redirected to the main functionality page of the application.

2.5.2 Registration

User can reach the registration screen, shown in Figure 6, by pressing the “Sign up” button located below the authentication form.

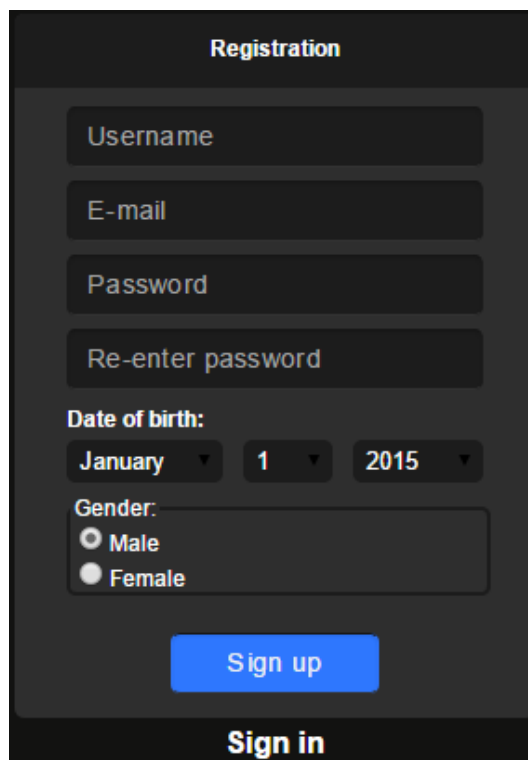
The image shows a registration form on a dark background. At the top, the word "Registration" is written in white. Below it are five input fields: "Username", "E-mail", "Password", "Re-enter password", and "Date of birth:". The "Date of birth:" field is composed of three dropdown menus showing "January", "1", and "2015". Below these is a "Gender:" section with two radio buttons labeled "Male" and "Female". At the bottom of the form is a blue "Sign up" button. Below the button, the words "Sign in" are written in white.

Figure 6: Registration form

For registration, user is required to choose and enter a valid username, which only contains letters, numbers and underscores and is no less than 4 characters long. Other characters are not allowed. The next field is email address after which comes the password field. The password must be at least 6 characters long and contain at least one number, one uppercase letter and one lowercase letter, to make the password more secure. User must enter the password twice. Both entries must match to make sure that the user did not misspell his password. For next, the user must enter his date of birth, which can be chosen from 3 dropdown lists: month, day and year. The last step in the registration is gender. Gender can be chosen by activating a button in front of the appropriate option. User is alerted if any of the information is not correct and user can then make the required corrections.

2.5.3 Application functionality

In the application page user is presented with a circular white area in which are the images. In the area images are placed circularly around an inner circle which we call the arena, which can also be seen in Figure 7. Images are generated by the algorithm and are taken from the database. Users can interact with the images by dragging them around and into the arena. Dragging is

triggered, when user hovers the mouse cursor over an image, presses down the left mouse button and moves the mouse around the screen while holding down the button. If user releases the mouse button the image is dropped. Hovering the mouse cursor over an image triggers another event which shows user a larger preview of the image. If user starts to drag the image, the preview disappears to give user better overview on where to drop the object.

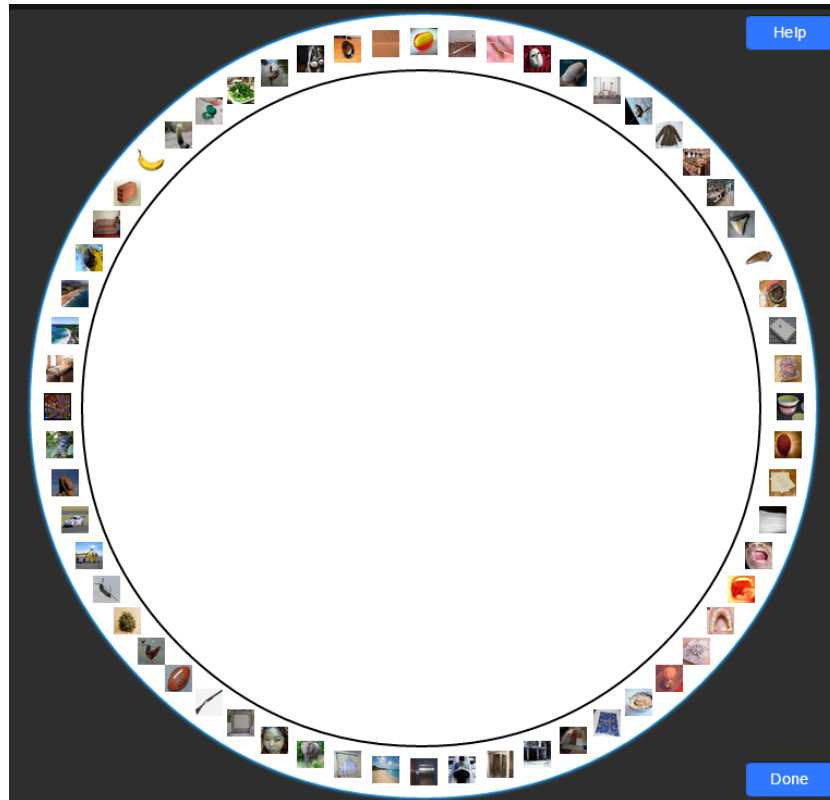


Figure 7: Arena surrounded by images

Images can only be dropped into the arena. The image returns to its previous location if the drag event is finished anywhere else than the arena. If the image is dropped into the arena it drops to the selected place.

User must drag all the images into the arena and place them based on their perceived similarity. Users' task is to compare the images based on their similarities and place the images more similar to each other close together and images that are less similar should be placed farther apart. User is strongly advised to use the whole arena. "Done" button can be clicked, after the user has finished comparing all the images. If the user clicks "Done" the data is processed and the information about the arrangement in the completed trial will be sent to the database and a new set of images will be generated for the user according to the algorithm described in Code and algorithmSection 2.3.2. After the information about the arrangement reaches the database

we estimate the RDM for the user according to the algorithm in Section 2.3.1, after the algorithm completes, the RDM is also sent to the database.

The help button is located in the upper right corner of the surroundings of the arena. Pressing that displays instructions, displayed in Figure 8. In it the user is explained what the task is and based on what criteria the images should be arranged.

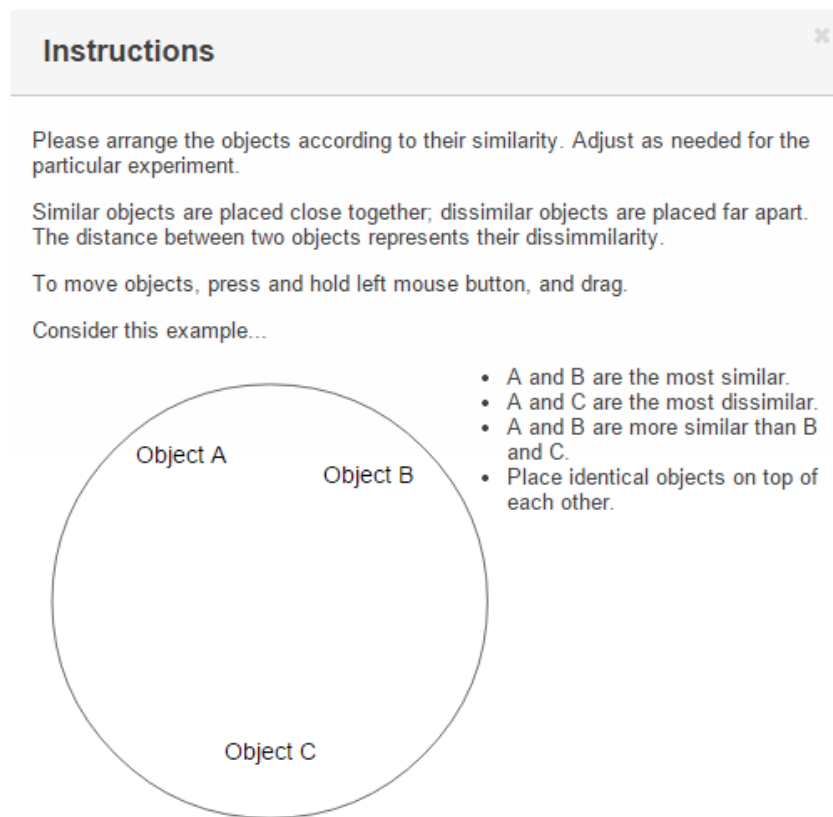


Figure 8: Instruction screen

User can end his session by pressing the “Log out” button located in the upper right corner of the page. After doing that the user will be redirected to the authentication page and the session will be terminated.

2.5.4 Accessing the resulting RDM

Above the authentication form there is a link, shown in Figure 9, to the RDM that is generated based on currently collected data. To access the RDM you do not have to register to the webpage. It is made easy for anyone to use the collected data for themselves as they see fit.

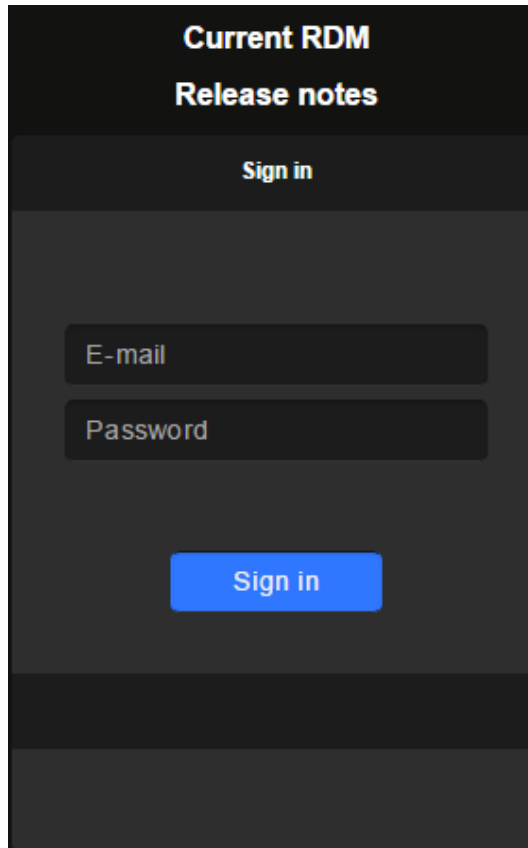


Figure 9: Link to the current RDM

User is redirected to another page after clicking the link. On that page the user is presented with the entire RDM upper triangle in the form of arrays. Each array there corresponds to a row of the representational dissimilarity matrix upper triangle. Values in the final RDM are arithmetic averages for each pair that we have collected data about, over all the user specific representational dissimilarity matrices.

2.6 Technologies and languages

2.6.1 PHP

PHP stands for PHP: Hypertext Preprocessor. PHP is a server-side scripting language, which is mainly designed for web-development. It is widely used all over the world and is compatible with most systems and servers. PHP code could easily be integrated with HTML code which

makes it perfect for web applications. [6] PHP was chosen for its flexibility, well written and freely available documentation, but also for its large community.

All the server-side communication in this application is done by using PHP. It is responsible for authentication, retrieving data from the database and sending data the application produces to the database. Most frequently queried information would be image and user data. The application main source of data comes from images comparisons, therefore image similarity data is most often sent to the database. PHP is also used to populate some forms in the application.

2.6.2 JavaScript

JavaScript is a dynamic programming language that is frequently used in different web-development [7].

JavaScript is used for implementing all of the algorithms, which are required to get similarity data from images. It calculates the distances between image elements and uses the results in different calculations. In addition the webpage utilises the popular JavaScript library called jQuery, which makes it easy to handle DOM elements and different events [13]. In the current application, jQuery is used to place images around the arena correctly. Additionally jQuery UI [9] elements are implemented to make it possible for users to drag images around the screen and drop them in the arena. This can be done by holding the left mouse button on the image and moving the image to the desired location.

Additionally, the webpage implements Ajax, also known as asynchronous JavaScript and XML. It is used in web-applications to send and retrieve data from the server asynchronously without interfering with the displayed page. [11]

2.6.3 MySQL

MySQL is a relational database management system. SQL stands for structured query language. MySQL finds many uses in web-application development and can be simply mixed into the PHP code. [14]

In this application, MySQL holds the responsibility of all data related actions. MySQL is used for different queries, which save data to the database as well as get the required data from it.

2.6.4 HTML

HTML which stands for HyperText Markup Language is markup language used for creating webpages. HTML divides into different elements that consists of tags. HTML elements form the structure of all websites. [10]

HTML is also used to form the building blocks of this website. It forms the structure of all pages in the application. Various HTML elements are implemented, such as forms, buttons and paragraphs. HTML provides a way to connect different elements of the application, for example the front-end code done in JavaScript and back end PHP code.

2.6.5 CSS

CSS which stands for Cascading Style Sheets is a stylesheet language that is used for editing the appearance of a document written in a markup language. It is mostly used in web-development to change the position and look of elements. [15]

In the current application, CSS is used to set the position and size of different HTML elements, as well as apply effects on mouse hover events. CSS also assigns different colours every element of the application. In addition it is used to edit text size and style.

3 Results

3.1 Subjects

Six people contributed to the current data collected from the webpage (age 22-41, 2 female). A total of 51 trials were made and the data from each trial was saved to the database.

3.2 Task

Subjects are presented with 60 images in the first trial. They are asked to drag the images to the arena. They can drag the images by clicking on an image with the left mouse button and while holding the button down and moving the mouse the images moves on the screen. When user releases the mouse button the image will be dropped and placed to the location the user specified. Users have to arrange all the images they are presented. Subjects are instructed to arrange images according to how similar or dissimilar they are. The image that user considers to be more similar should be positioned closer together and the images that the user thinks are less similar should be placed farther apart. Identical images should be placed on top of each other. The distance between two objects represents their dissimilarity.

3.3 Results summary

Resulting RDM was put together based on the collected data from a total of 51 trials. Before presenting the data the dissimilarity estimates were normalised to 1, so that the overall largest value would be 1. The other values were scaled accordingly. The RDM was generated by the web-application. The data was modified using a simple script, since the accessible RDM only contains the upper triangle of the whole matrix. For that reason we added the diagonal of zeros and the symmetrical other half using a small script. We also made the data suitable for MATLAB [16], which is used to produce the data visualisation.

In Figure 10 the entire RDM based on all of the data collected is visualised. Each square contains similarity data of the pairs it corresponds to. There is also a colour scale added for comparison to better evaluate the similarity. Dark blue represents most similar objects and red represents least similar objects. We have set the value of the pairs we have not yet collected data about to a default $\pi/10$. So far we have only collected data from 51 completed trials, which is also the reason most of the visualised RDM is light blue. X and Y axis represent the image id values.

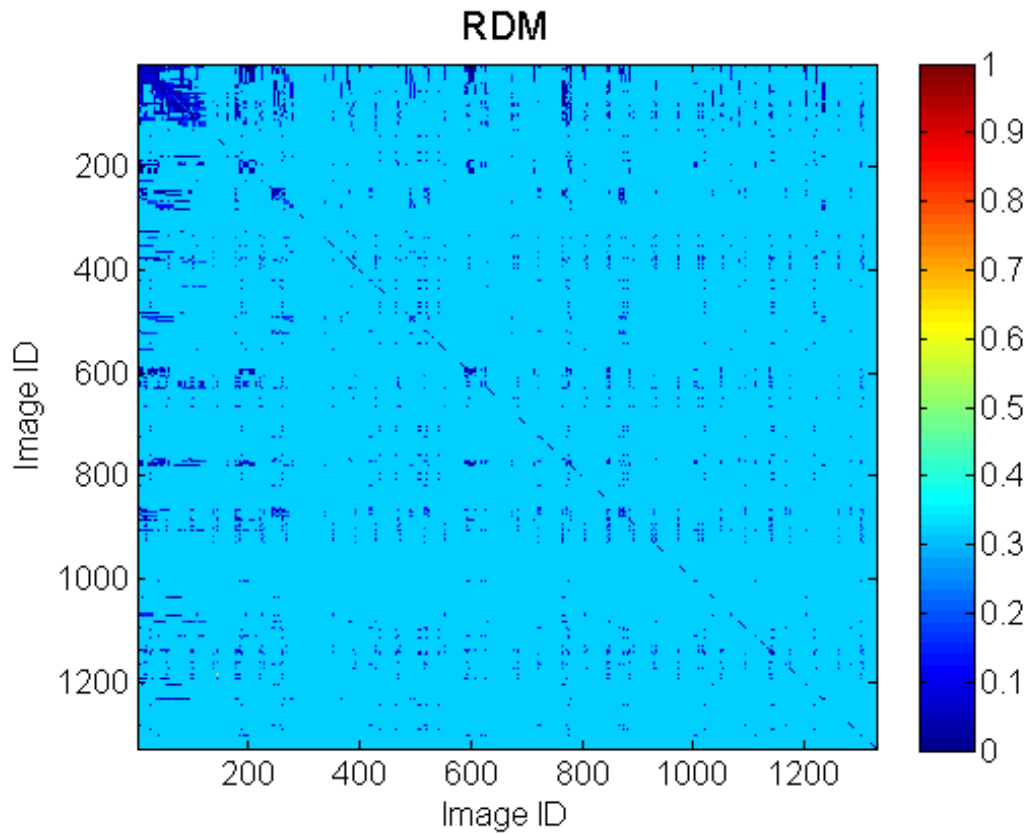


Figure 10: Visualisation of the entire RDM based on all of the images

In Figure 11 shows the RDM of a smaller subset of images to better show the similarities. Yet again the pairs we have not collected any data about are valued by default to $\pi/10$. A diagonal of zeros is clearly seen, shows that the same images are identical to each other. Each square shows the dissimilarity estimate of the image pair it corresponds to.

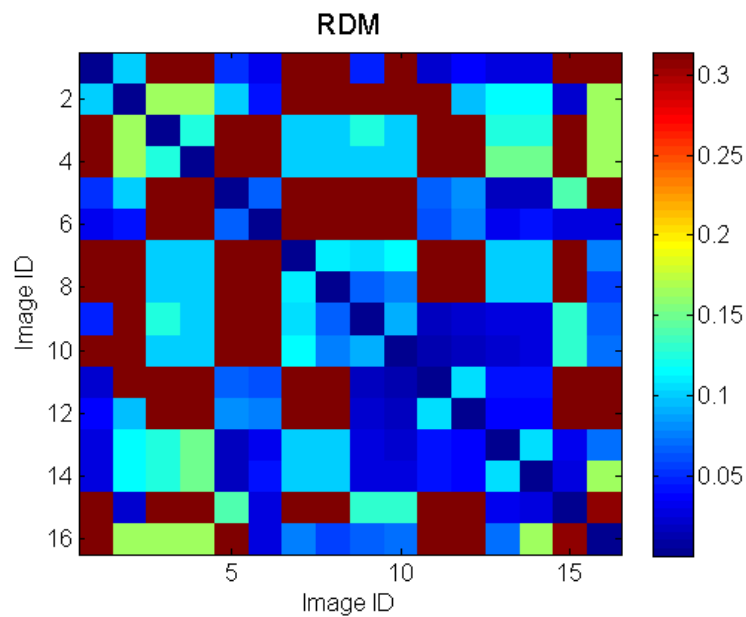


Figure 11: Visualised RDM of a smaller selection of items

Multidimensional scaling was used on the RDM with smaller selection of items and arrangement of these items was visualised into a 2D space, as shown in Figure 12. Distances between the images represent the similarities. Images that are far away from each other are less similar than the ones that are closer to each other. Blue dots show the precise location of the image and a small thumbnail is added to show which image each dot corresponds to.

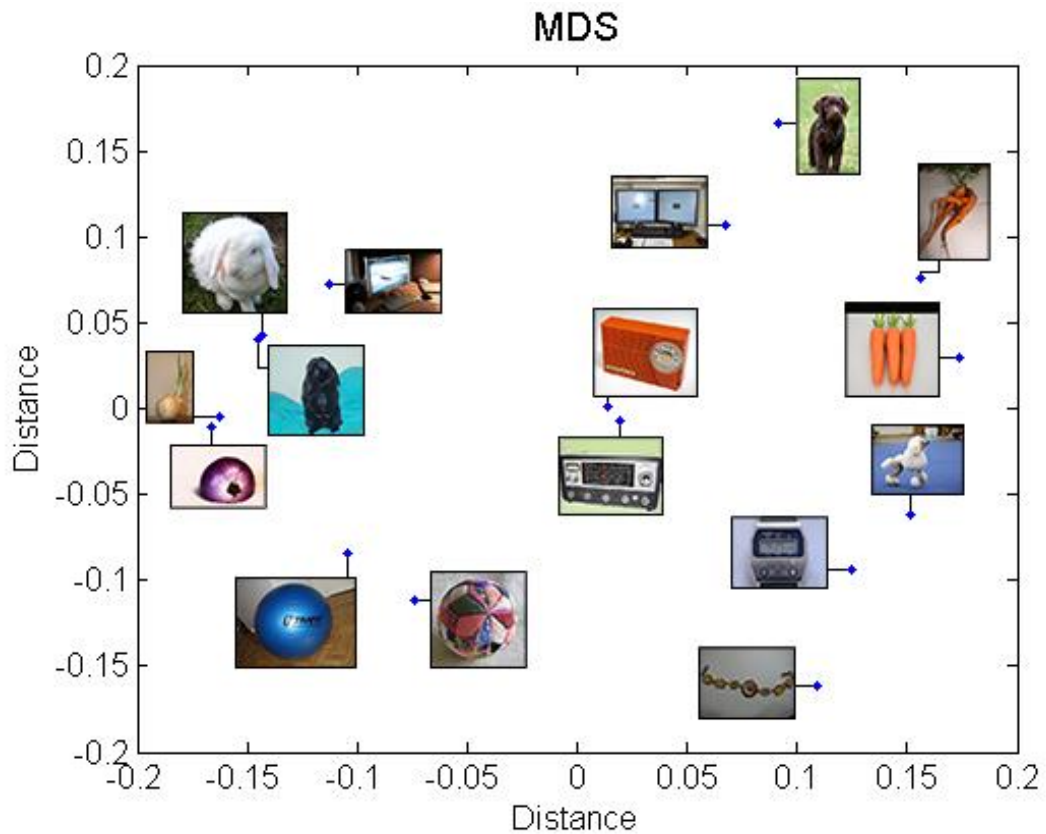


Figure 12: 2D visualisation of the RDM with the smaller section of items

Results in Figure 12 show that the images from the same category are rated as most similar to each other and placed close together in almost all of the cases.

More data would be needed to give a more accurate overview of the similarities and fill up the entire RDM shown in Figure 10.

4 Discussion

4.1 Necessity of the similarity ratings

So far the similarity ratings have only been collected for up to 100 images. The present web-application makes it possible to gain data more efficiently and conveniently. We increased the amount of images to around 1400 images, which could be arranged by people all over the world. Collecting similarity ratings at this big scale will result in a much more reliable dataset, which may allow researchers to compare the cognitive architecture of human vision to other measurements, for example neuronal measurements or artificial or machine visual systems. RSA provides a connection between these different systems. In addition it allows us to quantify and compare the data collected from these different sources. Knowing such a large representational dissimilarity matrix for human vision could potentially lead to improvements of artificial intelligence systems.

4.2 Webpage cons and pros

Using a webpage for collecting similarity ratings to generate representational similarity matrices is very convenient, because it can be easily accessed any time all over the world. This can in the future allow us to collect very large amounts of data which could result in a very reliable dataset that could be used in different studies. It also makes it possible for anyone to access the collected data.

The method of using a webpage for collecting scientific data has been used quite successfully before this project. One classical example would be the webpage made to map connections between retinal neurons in our brains, called EyeWire [17]. It is a web-application that has a game-like structure to motivate people to provide more data. This on its own shows how successful and efficient this “community based research” method can be.

4.2.1 Future improvements

The webpage developed here has some points that could use improvements. Firstly it is quite slow. This has to do with the very large matrices that need to be sent to the server, owing to the considerable number of images used. In addition the algorithms implemented take some time to process all the data that is presented to them. The algorithm tries to find the optimal subset to be presented to the user in the following trial, needs to go over all the pair, close to a million entries, and repeat this process multiple times to find the images for the next trial. Also the part of the code that generates representational dissimilarity matrices has a bottleneck. It takes time to go over each entry in the RDM.

The solution for these problems would be to rewrite the algorithms from the client side to the server side. This would require sending only information about each arrangement the user provides to the server and the server side code would process the data according to the algorithms. That would leave out repeatedly sending the representational dissimilarity matrices from client side to the server and vice versa, which would save a lot of time. In addition the algorithms could be further optimised to process only the fields that need to be taken into account. As an extra feature the application could send data to the server each time an image is dropped to the arena. If it is moved within the arena the data would just be updated. That would allow us to restore users' progress if something would happen in the middle of arranging the images.

Secondly the application would need a working mechanic to prevent cheating. It would raise the quality of the data collected if we could detect the arrangements that people inserted by randomly arranging the images for the sake of time efficiency. These arrangements would not provide us any useful information to improve the current dataset, it would reduce the overall accuracy of the result.

An anti-cheat system would never prevent all of the cases that would provide us inaccurate information, but it should decrease the number of cases when the incorrect data gets to the database. The system should process the arrangement the user inserted and compare the pairs in it to the average of the values we already have. If these values differ by a considerable amount on a large proportion of the pairs inserted we should add a marker for the user. If an account has reached the set specific limit of markers, the account should be removed from the system with all its data included. The system could also observe the amount of time a user spends on each arrangement. The time would correspond to the number of images in the arrangement. For example if the arrangement consists of only 10 images the average time to arrange all these images by giving each image a reasonable amount of thought process would be smaller than the time it takes to arrange 60 images. A marker should also be added to the user if all images are compared below a certain amount of time.

Additionally, there might be a potential drawback in which we cannot easily relate similarities between different people. It is possible that one person has a different similarity matrix than another. This leads to generalisations that may not exist in individuals. To overcome this problem a functionality could be implemented, which would allow to show individual users' representational similarity matrices separately from the RDM that includes the average values over all users.

Another feature that would make the process of registration smoother for people, would be alternative login systems, using popular social network sites. That would make it possible for people to avoid the tedious registration screen and the information would be automatically taken from their social network profile.

The application should also be better implemented to function as a game. Users should be given points according to the dissimilarity-evidence matrix. Points should be given for each pair that has a value greater than 0.5, which represents the end criteria the user should reach in order to finish. In addition a system for different achievements and goals should be presented. It would motivate users if they could compete with other people all over the world. The achievements should unlock users more features. For example users could unlock more images after they have completed certain arrangements or a certain image set. We can collect an even better dataset, if users are highly motivated to compare more images.

Lastly the web-application should be changed to work properly on a larger or a smaller screen. Currently the size of the arena, where the images are placed has a fixed size of 624 pixels to assure that all the similarity ratings are based on the same maximum size value. That means that on a smaller screen a tedious scroll bar appears, since the arena could not be fitted to the screen. Also on a much bigger screen the arena would be ridiculously small and much space that would ease the comparing process would be unused. The solution would be to make the arena size adapt according to the size of the screen. The arena size value used in the algorithms should be taken from the values of its attributes.

Conclusion

The goal of this thesis was to develop a web-application, which could be used by people around the world. The main functionality of the webpage was to allow users to compare images on their perceived similarity and place similar objects close to each other and less similar ones farther apart. This would generate a large dataset of image ratings from which a similarity matrix could be constructed.

The web-application was developed using different technologies and languages such as HTML, PHP, JavaScript and MySQL. The most difficult and time consuming part of the project was implementing and understanding the algorithms, which estimated the RDM and generated new subsets to the user. In addition 4 images for approximately 400 different categories were handpicked, so that the images would be as different as possible from each other.

The resulting web-application was used for collecting data, which is also presented in this thesis. Although there were not many subjects and only 51 arrangements were made, the application performed successfully and produced a representational dissimilarity matrix from the collected data. The collected data were also visualised.

The application can be further developed and some suggestions were made. In the future, if more data is collected it would produce a reliable dataset which could potentially be used to improve artificial intelligence algorithms designed to categorise objects.

REFERENCES

- [1] Jürgen Schmidhuber, "Deep Learning in Neural Networks: An Overview," 2014. [Online]. <http://arxiv.org/pdf/1404.7828.pdf> Last visited 13 May 2015.
- [2] Farzmahdi Amirhossein, Rajaei Karim, Ebrahimpour Reza, Khaligh-Razavi Seyed-Mahdi Ghodrati Masoud, "Feedforward object-vision models only tolerate small image variations compared to human," 2014. [Online]. <http://journal.frontiersin.org/article/10.3389/fncom.2014.00074/full> Last visited 13 May 2015.
- [3] Marieke Mur, Peter Bandettini Nikolaus Kriegeskorte, "Representational Similarity Analysis – Connecting the Branches of Systems Neuroscience," *Frontiers Research Foundation*, 2008. [Online]. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2605405/> Last visited 13 May 2015.
- [4] Marieke Mur Nikolaus Kriegeskorte, "Inverse MDS: Inferring Dissimilarity Structure from Multiple Item Arrangements," *Frontiers Research Foundation*, 2012. [Online]. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3404552/> Last visited 13 May 2015.
- [5] wikiHow. (2015) How to Create a Secure Login Script in PHP and MySQL. [Online]. <http://www.wikihow.com/Create-a-Secure-Login-Script-in-PHP-and-MySQL> Last visited 13 May 2015.
- [6] PHP. What can PHP do? [Online]. <http://php.net/manual/en/intro-whatcando.php> Last visited 13 May 2015.
- [7] Wikipedia Foundation Inc. (2015) JavaScript. [Online]. <http://en.wikipedia.org/wiki/JavaScript> Last visited 13 May 2015.
- [8] Greg Holt, Andrew Kepert Paul Johnston. (2009) JavaScript implementation of the Secure Hash Algorithm, SHA-512. [Online]. <http://pajhome.org.uk/crypt/md5/sha512.html> Last visited 13 May 2015.
- [9] The jQuery Foundation. JQuery user interface. [Online]. <https://jqueryui.com/> Last visited 13 May 2015.
- [10] Wikipedia Foundation Inc. (2015) HTML. [Online]. <http://en.wikipedia.org/wiki/HTML> Last visited 13 May 2015.
- [11] Wikipedia Foundation Inc. (2015) Ajax_(programming). [Online]. [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) Last visited 13 May 2015.
- [12] ImageNet. ImageNet. [Online]. [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) Last visited 13 May 2015.
- [13] The jQuery Foundation. (2015) jQuery. [Online]. <http://jquery.com/> Last visited 13 May 2015.
- [14] Oracle Corporation. (2015) MySQL. [Online]. <http://www.mysql.com/> Last visited 13 May 2015.

- [15] Wikipedia Foundation Inc. (2015) Cascading Style Sheets. [Online]. http://en.wikipedia.org/wiki/Cascading_Style_Sheets Last visited 13 May 2015.
- [16] The MathWorks, Inc. (2015) MathWorks. [Online]. <http://se.mathworks.com/products/matlab/> Last visited 13 May 2015.
- [17] EyeWire. EyeWire. [Online]. <http://blog.eyewire.org/about/>

LICENCE

Non-exclusive licence to reproduce thesis and make thesis public

I, Taavi Gilden (date of birth: 18th of April 1993),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1.reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2.make available to the public via the university's web environment, including via the DSpace digital archives, as of 26.06.2016 until expiry of the term of validity of the copyright,

Creating a Human Similarity Ratings Benchmark Database for Artificial Neural Networks,
supervised by Jaan Aru, Raul Vicente, Martin Hebart

2. I am aware of the fact that the author retains these rights.
3. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 14.05.2015